CS-308-2014 Final Report

Pool Bot

Team: Pool Off

TU - 10

Nikunj Saunshi – 100050007

Omkar Thakoor – 100050009

Diksha Meghwal – 100050039

Shina Johri - 100050048

Table of Contents

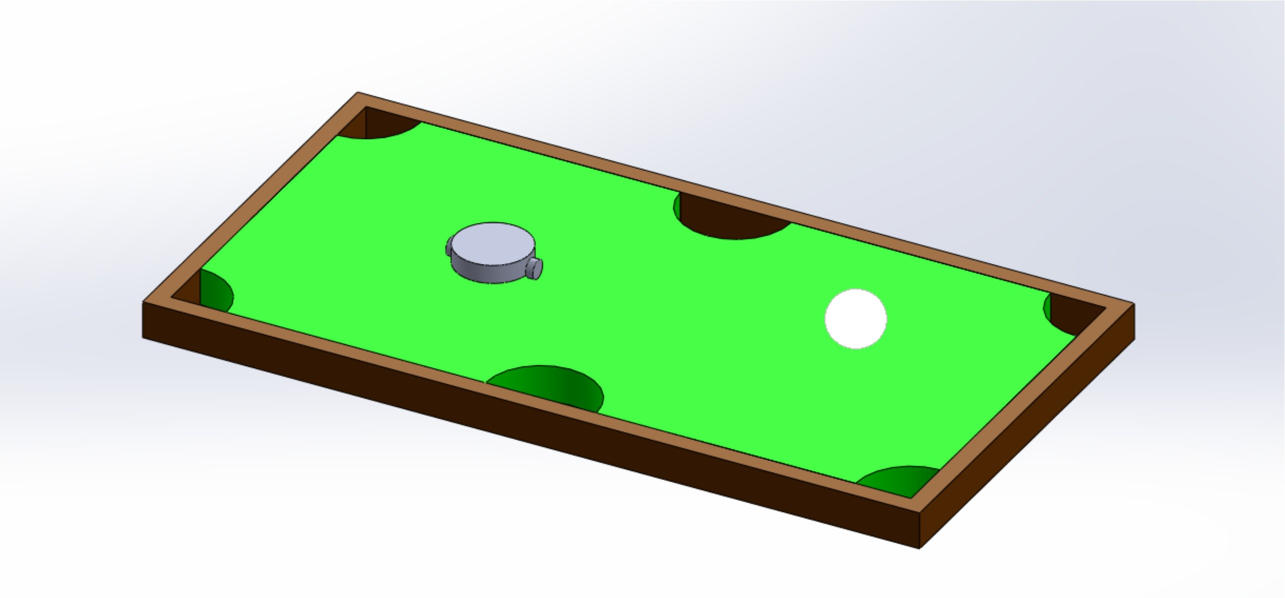
1. Introduction

We have designed a bot to automate the game of pool. In this project, we have implemented a simpler version of pool where we have just one ball in the arena and the bot is free to move anywhere in the arena. The aim is to pot the ball into a hole.

It is obviously useful to play pool. However, certain individual components of this project can be used for other applications. E.g. A similar impact mechanism can be used for games like carom. Also the image processing and positioning module can be used in numerous other applications

2. Problem Statement

We have a 165 x 110 cm arena with six holes, just like a standard pool table. We randomly place the ball and the bot in the arena. The bot needs to locate the ball and then find the optimal hole to pot the ball. It then needs to position and orient itself accordingly and then hit the ball into the hole. After the shot, the bot needs to be ready for the next shot.



3. Requirements

3.1 Functional Requirements

Image processing: An overhead fixed camera takes snapshots of the arena. This image is then used to first identify the individual components and then to find the centroid of the bot, ball and the holes. It then converts the pixel coordinates to real distances, and returns the coordinates of each component w.r.t to axes which are parallel to the table edges and the origin being one of the corner holes.

Positioning and orientation: Once the bot receives the coordinates of itself, the ball and the holes, it does the required calculations and finds the destination coordinates and orientation for itself to pot the ball into the optimal hole (which is the hole closest to the ball in this case). The bot then moves from the initial to the destination position and orients itself and is ready to hit.

Impact mechanism trigger: Once behind the ball, the bot triggers a signal and sets the stopper motor into motion. This enables the impact mechanism and the rod tries to hits the ball into the hole.

3.2 Non-Functional Requirements

Automation: Every aspect of the system needs to be automated. Once the ball and the bot are placed in the arena the user should not have to play any role in the successive steps.

Collision Avoidance: While orienting itself the bot should never collide with the ball.

Optimal shot: It should choose the hole so that the shot to be taken is least sensitive to any inaccuracies that may arise.

3.3 Hardware Requirements

FB5 bot: To move around the arena, do necessary calculations and trigger the impact mechanism.

USB Camera: To capture snapshots of the arena for determining the positions and orientation of the ball and bot.

Xigbee: For wireless communication between the laptop and the bot.

Motors: To control the stopper and the cue.

L293D Motor Driver: To automate the cue mechanism using signal inputs from the bot

3.4 Software Requirements

Matlab R2011a: For image processing and sending data via xigbee to the bot.

Keil uvision 4: To compile and build target files for Fire Bird 5

XCTU: To configure the XigBees

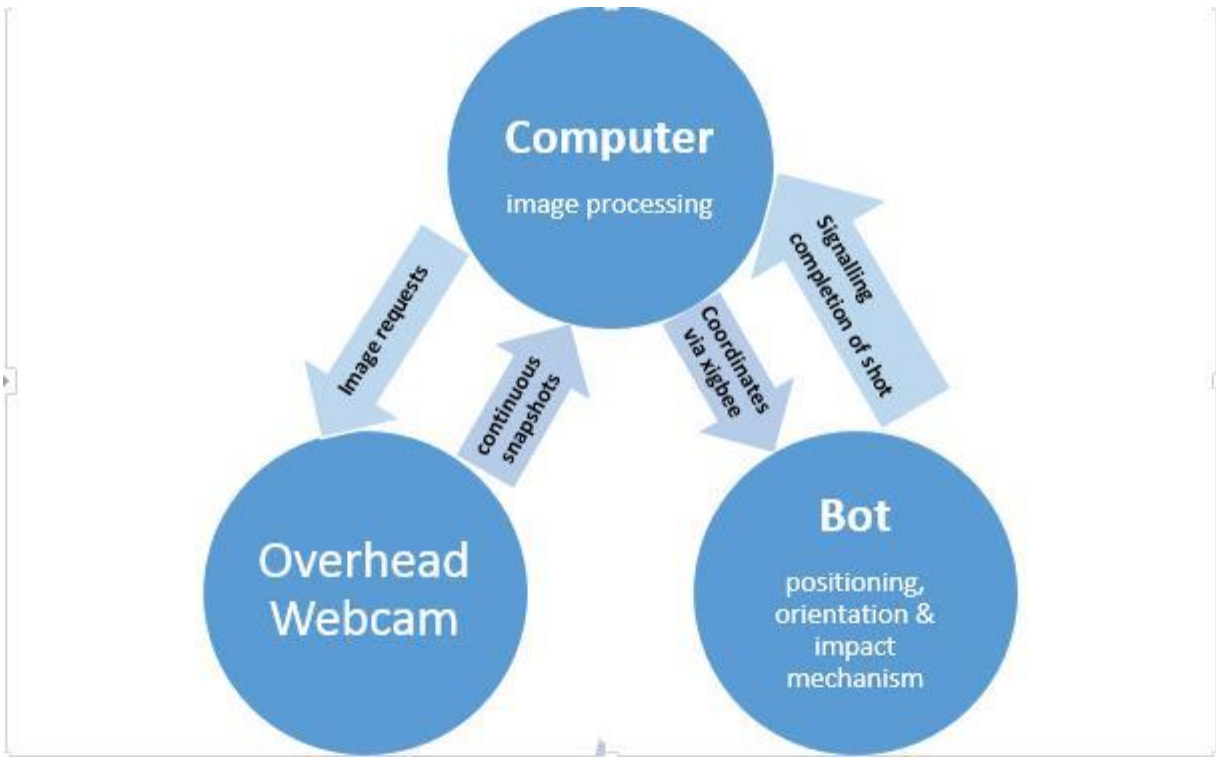
4. System Design

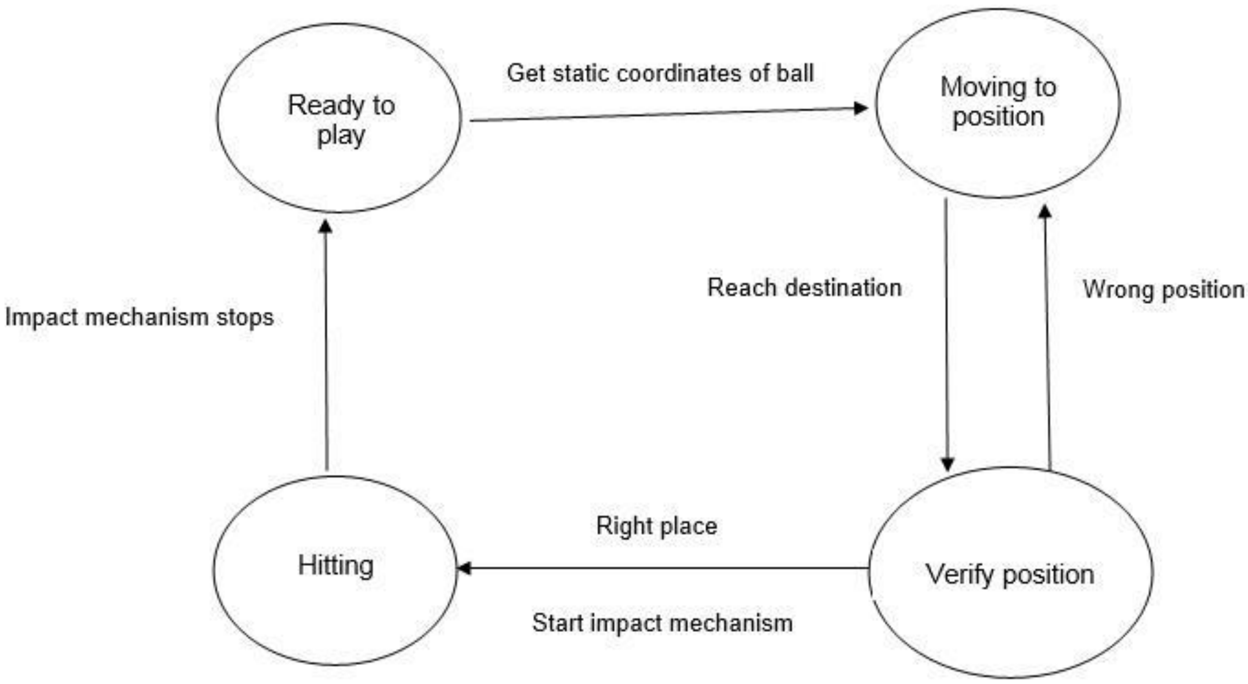
There is a fixed overhead USB camera connected to a laptop. This camera sends snapshots of the arena to the laptop.

The laptop then analyses the image, performs image processing and determines the coordinates of the ball, bot and holes. It scales these values to actual distances and sends these values to the bot. It uses xigbee for wireless transmission of these values

The bot uses these values sent by the laptop and determines the destination position to hit the ball into a hole. It makes an optimal choice of the hole and then orients itself accordingly.

The bot then triggers the impact mechanism to hit the ball into the hole.





FSM

5. Working of the System and Test results

Image Processing:

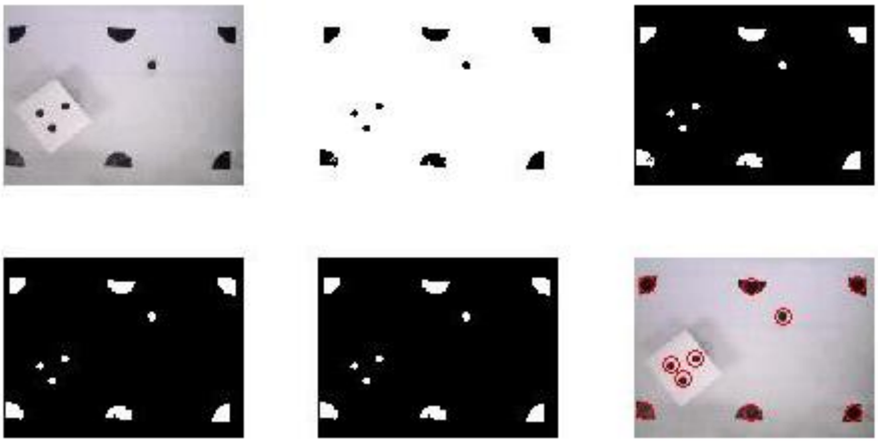
The image from the camera is first converted to a black and white image (any pixel above a certain threshold of luminosity is identified as white).

A Matlab function then finds continuous black regions and calculates the centroids for each black blob. These are pixel values for the centroids.

First a snapshot of the arena without the ball and bot (only with holes) is taken and the centroids obtained are stored as the coordinates of the holes. These coordinates are used to find the scaling factor between the pixel distances and the actual distances. Within 10 sec, the bot and the ball need to be placed randomly placed in the arena.

A snapshot is now taken and 4 new coordinates are obtained (3 for the bot and 1 for the ball) as shown in the previous figures.

These are converted to actual distances using a scaling factor obtained earlier.



Wireless communication:

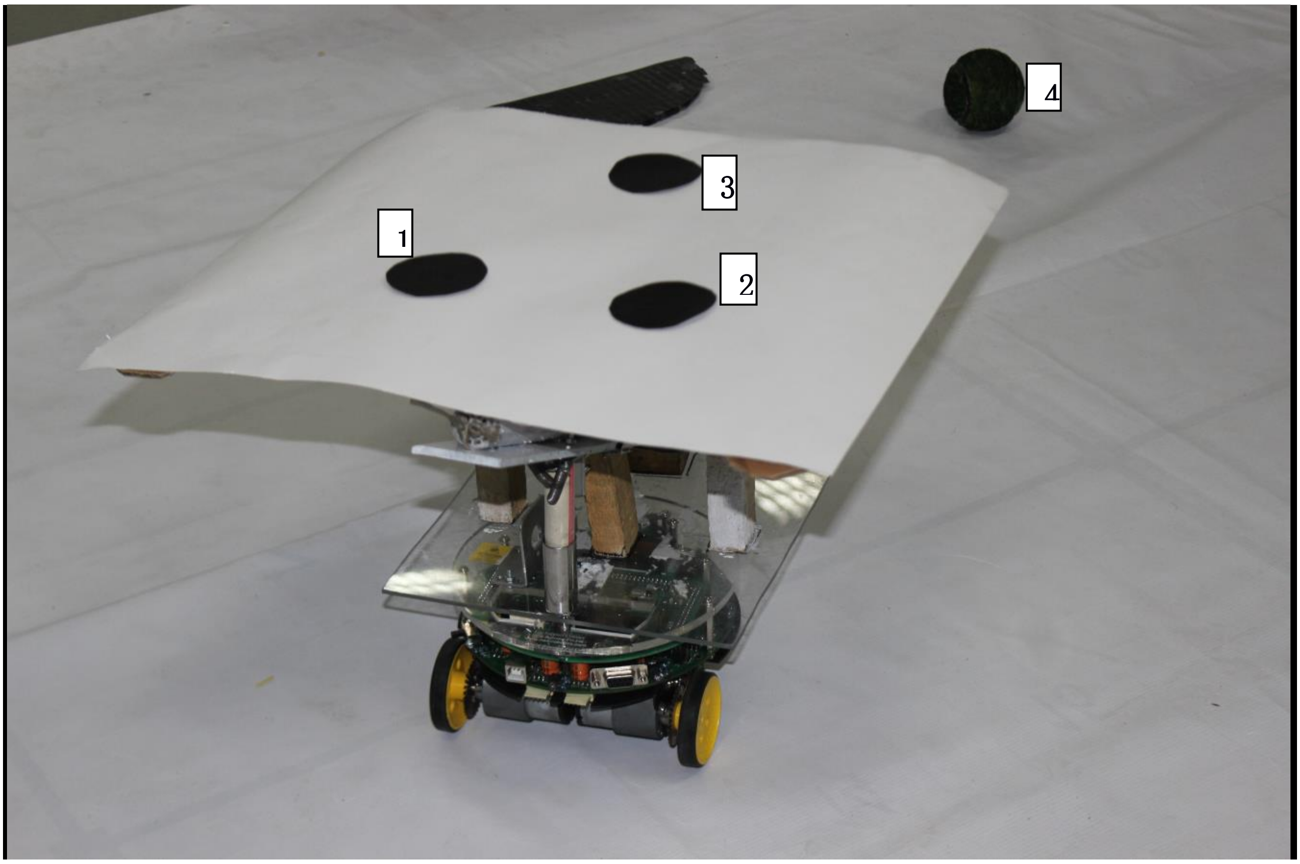
These 4 coordinates are sent in mm and each coordinate value is sent as a stream of 4 characters (digits) using xigbee. These values are re-chunked and stored as integers, where each value is in mm.

Component detection:

The closest 2 coordinates among the 4 are identified as the circles 1 and 2 of the bot. These 2 circles give the pivot coordinates for the bot.

The next closest coordinate is the circle 3 of the bot. This, along with the previous 2 coordinates, gives the angle of the bot with the horizontal axis.

The 4th coordinate is now assigned to the ball.



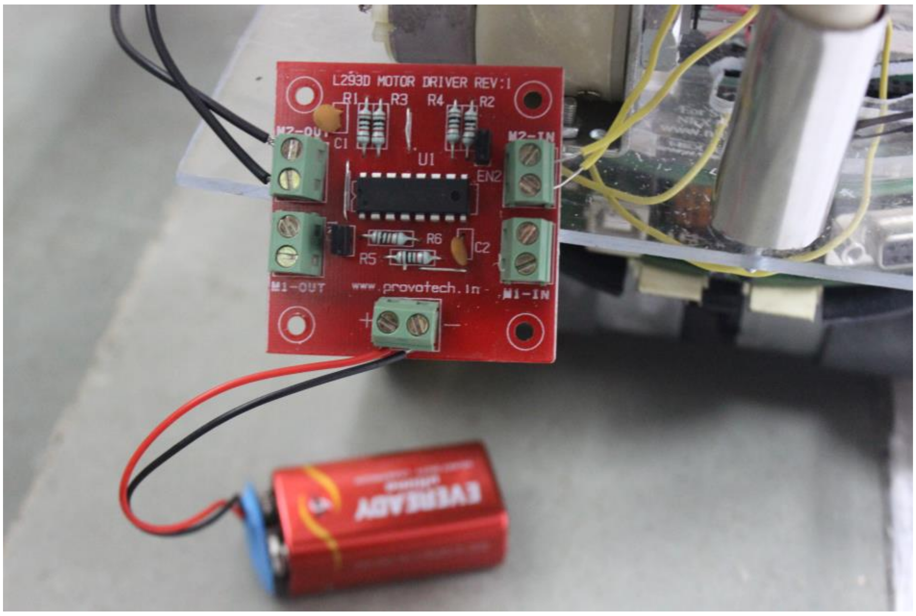
Positioning and Orientation:

These coordinates are used to determine the hole closest to the ball and the destination coordinates and angle of the bot for the appropriate shot.

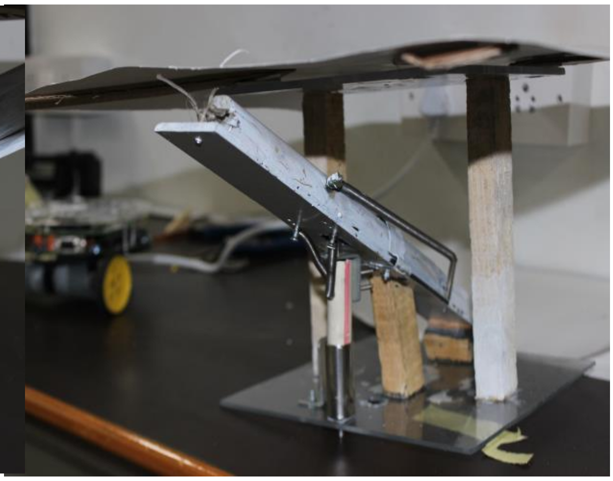
The bot then uses the wheel encoders to move itself by the required distance, first along the x-axis and then along the y-axis, and then rotate the appropriate angle so that all that is left would be to hit the ball.

Impact Mechanism:

Once the bot reaches the destination, it triggers the impact mechanism. Pins from the left shaft are used as signal inputs to automate the triggering.



The impact mechanism is basically a rod that has been held in a position by an extended spring with the help of a stopper. When we trigger the impact mechanism we basically start the motor that pulls down the stopper. As soon as the rod is released from the stopper, it gives a sudden jerk in the forward direction hitting the ball. After the impact has been created, the rod is pulled backwards with the help of another motor. The stopper is also released by the motor and is forced upwards with the help of springs. When the rod is pulled back to its original position its again gets locked by the stopper.

Testing:

Module	Methods for testing	Results
Image Processing	- Bot and ball were placed at random positions and their coordinates were obtained from image processing - These coordinates were then compared with the actual distances of the objects from the origin	- The coordinates obtained were within a range of 1 cm from the actual coordinates - The scaling factor was adjusted in order to obtain more precise results
Positioning and Orientation	- The initial and starting positions and orientations were hard coded and given as an input - This was tested by checking if the bot actually ends up at the destination position and orientation	- The final position was within a range of 5 cm from the desired destination and angle within a range of 5 degrees - This is mainly because of the inertia of the bot and inaccuracies in the wheel encoders - Offsets were introduces in the distances and angles to take care of these inaccuracies
Impact Mechanism	- The ball was placed right in between the bot and a hole and the impact mechanism was triggered. - The rotations of the motor controlling the stopper was the most important factor here	- This module worked perfectly fine. - The only problem that could arise is the discharging of the battery as time progresses. This would require the motor to rotate for more time

6. Discussion of System

a) What all components of your project worked as per plan?

- We proposed to calculate the position of the balls, bot and the holes with the help of an overhead camera and we have been able to do so with quite a precision.
- We have implemented an algorithm to decide the optimal hole to pot the ball.
- We have been able to do the positioning and orientation of the bot pretty precisely for the intended shot.
- We have designed and implemented an impact mechanism that is able to strike the ball with a jerk very similar to an actual pool shot.

b) What we added more than discussed in SRS?

- Nothing additional has been added that what had been proposed in the SRS.

c) Changes made in plan from SRS:

- We came up with a completely different design for the impact mechanism. The mechanism mentioned earlier used two rods to provide a to-n-fro the motion of the rod in a straight line. However, this method didn't provide enough thrust for a good shot. In our present mechanism, we have used extended springs to provide the thrust for the shot and have used a restrictor to restrict the motion of the rod in a straight line.
- We have not been to automatically restart the bot for another shot.

7. Future Work

- This game currently has only a single ball. We can introduce more no of balls. This will require tackling issues like collision-detection and better image processing.
- In such a scenario, we can place the bot outside the arena to combat collision but this will require the bot to use a longer cue in the impact mechanism.
- We can also introduce better shots like rebound shots, cut shots and relay shots.
- We can use the impact mechanism and the positioning and orientation scheme in other similar sports like carom.
- The image processing and the positioning and orientation of the bot can be useful in many applications that require precise motion of the bot.

8. Conclusions

- The major hurdles in this project were the impact mechanism, precision of image processing and perfect motion of the bot
- In real life scenarios, most of these are difficult to achieve. For e.g. a mismatch between the two wheels in the bot may lead to inaccuracies
- Our final project does not always hit a perfect shot due to some issues w.r.t. the points mentioned above
- However there is scope for improvement, and with more time and resources, the system can